

MARWIS: A Management Platform for Heterogeneous Wireless Sensor Networks

by Gerald Wagenknecht, Markus Anwander and Torsten Braun

A heterogeneous wireless sensor network (WSN) contains different types of sensor nodes. To operate such a WSN, we present MARWIS (Management ARchitecture for Wireless Sensor Networks). It uses a wireless mesh network as a backbone and offers mechanisms for visualization, monitoring, reconfiguration and updating program code.

A wireless sensor network (WSN) may run different applications for different tasks, such as event detection, localization, tracking, or monitoring. Different types of sensor node are therefore required, and to handle heterogeneous WSNs with a large number of these different sensor nodes, a comprehensive management architecture is also necessary. We present MARWIS, a Management Architecture for heterogeneous Wireless Sensor Networks, which supports common management tasks such as visualization, monitoring, (re)configuration, updating and reprogramming. It takes into account the specific characteristics of WSNs and the restricted physical resources of the sensor nodes. These include battery life, computing power, memory, network bandwidth and link quality.

One of the main features of MARWIS is its hierarchical architecture. We divide a large heterogeneous WSN into smaller sub-networks, each of which contains sensor nodes of one specific type. A

wireless mesh network (WMN) operates as the backbone and builds the communication gateway between these sensor sub-networks, the WSN and the Internet. Wireless mesh nodes perform the management tasks, and are controlled by a management station located in the Internet. A possible scenario is shown in Figure 1.

The use of a hierarchical architecture has various advantages. Sensor nodes, which are normally unable to communicate with each other due to incompatible radio chips, can be interconnected using wireless mesh nodes. Furthermore, dividing a huge WSN into smaller sensor sub-networks decreases the number of hops required to reach each sensor node. Specifically, each sensor node reaches the next wireless mesh node (which is the communication gateway) within three to four hops. This results in better communication performance with a lower round-trip time, lower jitter and less packet loss. A further advantage of using a WMN is that a new sensor node

platform can be easily integrated into the heterogeneous WSN.

The architecture used to manage heterogeneous WSNs efficiently contains the following structural elements: one or more management stations, several mesh nodes as management nodes, sensor node gateways plugged into a wireless mesh node, and the heterogeneous sensor nodes. The management functionality is placed on the wireless mesh nodes, meaning the resource-limited sensor nodes have fewer management functions to perform, which in turn reduces memory and computation requirements. A user can perform management tasks using a management station, and this can be remotely located on the Internet.

Using a graphical user interface, the topology of the heterogeneous WSN with all the sensor sub-networks is visualized. The status information about every sensor node is monitored and displayed. This includes hardware features

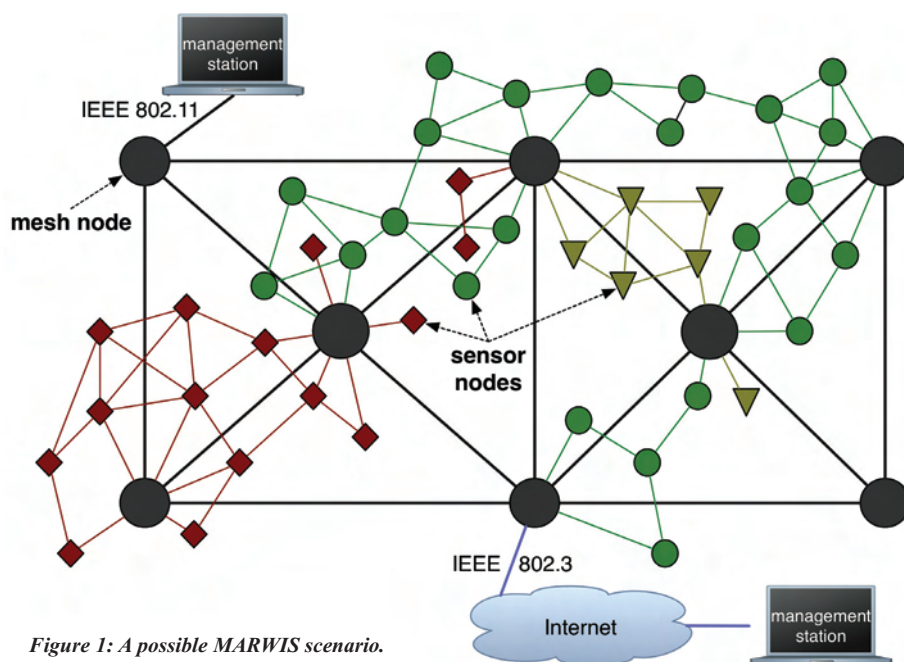
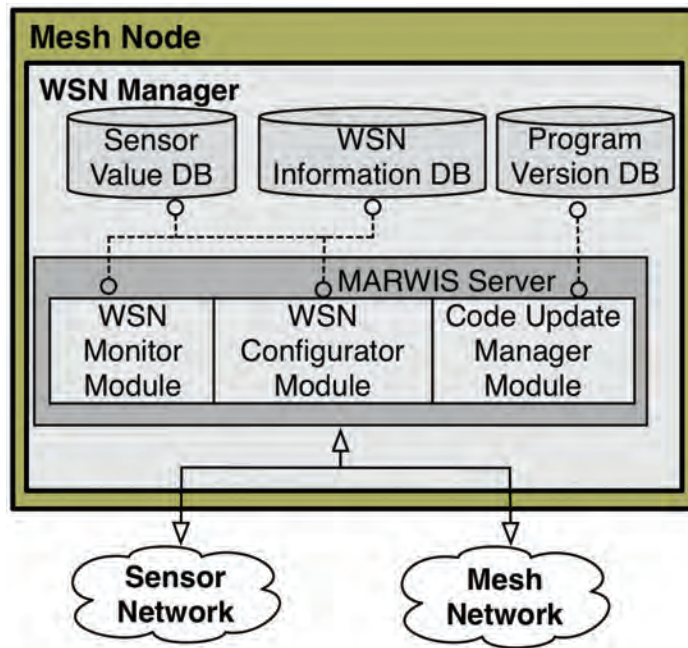


Figure 1: A possible MARWIS scenario.

Figure 2: The WSN manager provides the management functionality for the different sensor sub-networks. It consists of three databases and the MARWIS server with three modules.



(micro-controller, memory, transceiver), software details (operating system versions, protocols, applications), dynamic properties (battery, free memory) and, if available, geographical position information. The applications running on the sensor nodes or network properties can be reconfigured using the user interface. Furthermore, updating and reprogramming the sensor nodes is a very important issue. In large WSNs manual execution of this task is unfeasible, and a mechanism to handle it automatically and dynamically over the network is required. Both the operating system and applications must be updated, either fully or partially.

The WSN manager located on the mesh nodes provides the management functionality for the different sensor sub-networks. It consists of three databases and the MARWIS server with three modules, as shown in Figure 2.

The WSN information database stores all information about the sensor nodes and the WSN, such as the topology (neighbours, addresses) and states of the sensor nodes (battery, memory). The program version database stores all versions of all programs for all platforms, which can be installed in the sensor nodes. Finally, the sensor value database stores all data measured by the sensors. To get information about the relevant mesh node are queried. This means a direct connection to the sensor node is unnecessary, which saves time

and energy. If newer information is required, the sensor node can be queried directly.

The MARWIS server contains three modules for the management tasks. The WSN monitor module connects to the WSN information database and to the sensor value database in order to handle requests from the management station. It also stores data coming from the sensor nodes into the databases. The WSN configurator module is responsible for the configuration tasks. It queries properties from the sensor nodes and stores them in the WSN information database. The code update manager module stores newly received program images (and related information) in the program version database and notifies the management station about available programs.

The Sensor Node agent is the complement of the MARWIS server and performs the management tasks on the sensor nodes after message exchange with the MARWIS server.

The architecture is currently being implemented and tested in a small real-world testbed. A small Linux distribution (kernel 2.6.14.6) is running on the mesh nodes; the MARWIS server is being implemented in C using sockets; the databases are managed with MySQL; the API for accessing the databases is implemented in C; and Contiki is running on the sensor nodes as the operating system.

Link:

<http://www.iam.unibe.ch/~rvs/research/mancom.html>

Please contact:

Gerald Wagenknecht
University of Bern, Switzerland
Tel: +41 31 511 26 36
E-mail: wagen@iam.unibe.ch

Markus Anwander
University of Bern, Switzerland
Tel: +41 31 511 26 34
E-mail: anwander@iam.unibe.ch

Torsten Braun
University of Bern, Switzerland
Tel: +41 31 511 26 31
E-mail: braun@iam.unibe.ch